Name: Tim Blazytko

E-Mail: mr.phrazer@gmail.com

PGP Key ID: 0xE41B49AD

Datum: 14. Juni 2008

Thema: Lokale und LAN-interne Angriffsszenarien auf
Microsoft Windows NT 5.0-, 5.1- und 5.2-Systeme

Inhaltsverzeichnis

Abbildungsverzeichnis

1. Einleitung

Microsoft Windows NT 5.0-, 5.1- und 5.2-Betriebssysteme sind weit verbreitet und werden für die unterschiedlichsten Anwendungen genutzt, sowohl von Privatanwendern als auch von Firmen, als Arbeitsmittel für den Anwender oder als Server für ein Unternehmen. Ein Betriebssystem sollte einen bestimmten Grad an Sicherheit gewährleisten. Es sollte zum Beispiel über eine sichere Authentifizierung verfügen, das heißt, dass unter anderem eine vernünftige Passwortsicherheit existieren sollte.

Deshalb werden in dieser Arbeit Angriffsszenarien vorgestellt, welche ohne Exploits - Software oder Sequenzen von Befehlen, mit welchen man spezifische Sicherheitslücken bzw. Fehlfunktionen ausnutzen kann - auskommen, das heißt, dass Angriffsszenarien vorgestellt werden, welche Designschwächen in der Betriebssystem- und Netzwerkarchitektur von Windows NT 5 ausnutzen. Diese zwei Bereiche werden separat vorgestellt. Im ersten Teil wird zunächst der theoretische Hintergrund zur Systemarchitektur von Windows NT 5-Systemen vermittelt, anschließend die Funktion von Hashalgorithmen im Allgemeinen erklärt und im Speziellen auf die Hashfunktionen, welche die erwähnten Systeme verwenden, eingegangen. Danach folgt eine Beschreibung zum Auslesen der Hashes, eine Ausführung der Schwächen der bei Windows implementierten Hashfunktionen und ein Angriff auf diese mittels rainbow tables.

Im zweiten Teil wird der theoretische Ansatz der Funktionsweise der Netzwerkkommunikation anhand des OSI-Schichtenmodells vorgestellt. Es folgt eine Beschreibung der Funktion und der Windows-internen Verarbeitung der MAC-Adresse sowie der Möglichkeit, diese zu fälschen und der möglichen Gefahren im LAN (Local Area Network), welche durch diese entstehen. Außerdem wird die Funktionsweise des Internetprotokolls der Version 4 angeschnitten, eine Möglichkeit der Fälschung der IP-Adressen vorgestellt und ebenfalls auf entstehende Risiken hingewiesen. Mit den gewonnen Erkenntnissen folgt eine ausführliche Beschreibung des ARP-Protokolls, dessen Aufbau und Funktion. Zwei Arten der ARP-Paket-Manipulation sowie die Gefahren und Angriffsmöglichkeiten, welche sich daraus ergeben, werden abgehandelt und die Funktionsweise eines Sniffer, welcher das Mitlesen des Datenverkehrs ermöglicht, erklärt.
Am Schluss jedes Teils werden mögliche Schutzmaßnahmen gegen die jeweiligen Angriffe beschrieben.
Am Ende werden sämtliche Forschungen zusammengefasst und abgewägt.

2. Lokale Angriffsszenarien auf Microsoft Windows NT 5-Systeme

2.1 Architektur der Windows NT 5-Systeme

2.1.1 Windows NT 5 Systemarchitektur

Ein wichtiger Bestandteil der Architektur der Betriebssysteme Windows 2000 (Version NT 5.0), Windows XP (Version 5.1) und Windows Server 2003 (NT 5.2) der Firma Microsoft ist die Windows-Registrierungsdatenbank (Registry). Für die folgenden Untersuchungen wird lediglich der Registry-Pfad HKEY_LOCAL_MACHINE benötigt, welcher in den Dateien im Ordner *%windir%\System32\Config*, genauer in den Dateien SAM (Security Account Manager), SECURITY, software und system, gespeichert ist. Im Rahmen der weiteren Forschungen sind einzig die Dateien SAM und SECURITY, welche die sicherheitsrelevanten Einstellungen, wie lokale Benutzerkonten, deren Gruppen, Berechtigungen sowie Systemrechte beinhalten, und system, welche die beim Systemstart benötigten Hardware-Informationen, wie zum Beispiel Gerätetreiber, beinhaltet, von Bedeutung . In der SAM-Datei sind alle Benutzer und Passwörter in durch Hashfunktionen verschlüsselter Form abgelegt.[1]

2.1.2 Hashfunktionen

Eine Hashfunktion ist eine mathematische Funktion, in welcher eine unbestimmte Menge X auf eine endliche Menge Z abgebildet wird. Die Hashfunktion h(x) muss für jeden Input x (x ist Element von X) anwendbar sein und der Output von h(x) muss immer die gleiche Länge haben. Außerdem sollten diese Funktionen effizient (schnell berechenbar und leicht zu implementieren) und chaotisch (eine geringfügige Veränderung des Inputs soll eine enorme Änderung des Outputs als Folge haben) sein. Weil Hashfunktionen im Bereich der Kryptographie zum Beispiel für die chiffrierte Speicherung der Passwörter in einer Datenbank bedeutsam sind, müssen sie bestimmte Kriterien erfüllen: Sie sollten eine Einwegfunktion sein, das heißt, dass es praktisch unmöglich sein sollte, zu einem Ausgabewert h(x)=z einen Eingabewert x zu finden, und sie sollten kollisionsresistent sein, das heißt, dass es praktisch unmöglich sein sollte, sowohl zu einem gegebenen Eingabewert x ein davon verschiedenes x' zu finden (weak collision resistance), als auch zwei unterschiedliche, zufällige Eingabewerte x und x' zu finden (strong collision resistance), so dass jeweils h(x)=h(x')=z gilt, zwei Eingabewerte den gleichen Ausgabewert haben. Die Erfüllung des dritten Kriteriums beinhaltet die Erfüllung der anderen Kriterien.[2] Es gibt zwei Wege, um Hashfunktionen anzugreifen: Die collision attack und die preimage attack. Bei der ersten Variante wird versucht, zu zwei beliebigen Inputs einen gleichen Output zu finden (vgl. Kriterium 3), bei der zweiten Variante wird versucht, zu einem bestimmten Input einen weiteren Input zu finden (vgl. Kriterium 2), so dass beide den gleichen Output haben. Auf Grund des Geburtstagsparadoxons ist die Wahrscheinlichkeit der Durchführung einer erfolgreichen preimage attack wesentlich geringer.[3]

---

1 Für detailliertere Beschreibungen vgl. Microsoft (2006a).

2 Vgl. Bleikertz/Bugiel (2007), S. 20-23.

3 Zum besseren Verständnis der drei Kriterien vgl. eventuell mit dem „Geburtstagsparadoxon“, Wikipedia (2008a).

2.1.3 Hashfunktionen bei Windows NT 5–Systemen

Windows NT 5–Systeme haben zwei Hashfunktionen implementiert, die LM (LAN-Manager)- und die NTLM (NT LAN-Manager)-Hashfunktion, welche beide standardmäßig aktiviert sind. Der LM–Hash ist lediglich aus Gründen der Abwärtskompatibilität zu früheren Microsoft Windows-Betriebssystemen implementiert.[4] Der LM-Hash weist bei seiner Berechnung einige Schwächen auf: Erstens wird jeder Buchstabe des eingegebenen Passwortes in einen Großbuchstaben transformiert. Zweitens wird dieses Passwort mit Nullen gefüllt oder gekürzt, damit der String 14 Byte groß ist. Drittens wird das Passwort in zwei 7-Byte-Hälften aufgeteilt und jede Hälfte für sich selbst gehasht.[5]

Der NTML-Hash weist einige Verbesserungen auf: Er verwendet als Zeichensatz den Unicode, ermöglicht folglich die Benutzung einer größeren Anzahl an Zeichen und verzichtet auf die Umwandlung in Großbuchstaben. Des Weiteren teilt er den Input nicht auf, sondern hasht ihn vollständig, so dass der Aufwand bei einem starken Passwort bei einer brute force attack wesentlich höher ist. Ähnlich der LM-Hashfunktion wird das Passwort mit Nullen gefüllt oder gekürzt, damit der String 16 Byte groß ist.[6]

2.2 Lokale Angriffsszenarien

2.2.1 Getting the Hash

Ein Zugriff auf die in der SAM-Datei abgelegten Hashes ist während der Laufzeit des Betriebssystems Windows, wegen der Verwendung dieser durch das System, genauer, durch die Datei lsass.exe aus dem Ordner *system32*, wodurch für alle Benutzergruppen jegliche Zugriffsmöglichkeiten und -berichtigungen entfallen, nicht möglich.[7]

Weil die SAM-Datei nicht in Gebrauch ist, wenn es Windows nicht ist, ist es möglich, durch das Booten eines anderen Betriebssystems, zum Beispiel eines anderen installierten oder eines auf einer Live-CD, auf das Dateisystem des Zielsystems zuzugreifen und die SAM zu kopieren bzw. auszulesen und aus ihr die Hashes zu exportieren.

Seit dem Service Pack 3 für Windows NT 4.0-Systeme hat Microsoft eine Datei namens syskey.exe in dem Ordner *system32* implementiert. Laut Microsoft „bietet [diese Datei] starke Verschlüsselung der Kennwortinformationen in der Registrierung“[8], das heißt, dass ein Auslesen der SAM nicht zum Erfolg führe, weil die Hashes zusätzlich mit dem system key (auch: bootkey) decodiert werden müssten. Außerdem sagt Microsoft, dass der „Systemschlüssel unter Verwendung eines komplexen Verschlüsselungsalgorithmus im lokalen System gespeichert wird“.[9] Daraus kann man schließen, dass man den Systemschlüssel auslesen kann, sofern man weiß, wie er berechnet wird bzw. wo er abgelegt ist, und anschließend die aus der SAM exportierten Inhalte entschlüsseln kann (security by obscurity). Bei detaillierterer Betrachtung des Algorithmus erkennt man, dass die „starke Verschlüsselung“ einzig eine Permutation aus vier speziellen Registry-

4 Vgl. Microsoft (2004a).
5 Für eine ausführliche Berechnung des LM-Hashs vgl. Glass (2006), theLmResponse.
6 Für eine ausführliche Berechnung des NTLM-Hashs vgl. Glass (2006), theNtlmResponse.
7 Für eine detaillierte Beschreibung des Bootvorgangs der Windows NT 5–Systeme vgl. Microsoft (2005a).
8 S. Microsoft (2004b).
9 S. Microsoft (2004b).

Schlüsseln ist.[10] Das Programm Bkhive von Nicola Cuomo greift auf die Datei system (im gleichen Ordner wie SAM) zu, liest aus ihr die Schlüssel aus der Registrierungsdatenbank aus und vertauscht diese, bis es den bootkey berechnet hat und gibt ihn aus.

Bei uneingeschränktem Zugriff auf die Dateien SAM und system kann man mit dem Programm Samdump2, ebenfalls von Nicola Cuomo geschrieben, immer die Benutzernamen und Passwort-Hashes aus der SAM exportieren, weil es, falls die Hashes mit dem system key decodiert sind, den implementierten Algorithmus von Bkhive ausführt, die Hashes decodiert und ausgibt.[11]

Es gibt auch eine Möglichkeit, die Inhalte der SAM zur Laufzeit zu exportieren. Eine ist zum Beispiel, die Hashes aus dem Arbeitsspeicher auszulesen und auszugeben. Dies ist möglich, weil Windows die SAM zur Laufzeit in Benutzung hat und bei der Authentifizierung des Benutzers das eingegebene Passwort gehasht und mit dem Inhalt der SAM verglichen, somit der Inhalt der SAM im RAM (Random Access Memory) gespeichert wird. Für den Zugriff auf diesen Teil des Arbeitsspeichers sind administrative Rechte erforderlich. Ist ein Benutzerkonto nicht nur lokal, sondern auf der gesamten Domäne mit administrativen Rechten ausgestattet, hat es Zugriff auf den Speicherbereich, in dem die Hashes abgelegt sind, jedes einzelnen Computers innerhalb dieser Domäne. Das Programm PwDump4 des Autors bingle führt diese Schritte aus, wobei beim Auslesen ähnlich wie bei dem Programm Samdump2 vorgegangen wird, bis auf die Ausnahme, dass nicht der Speicherort der SAM angegeben und ausgelesen, sondern auf den Inhalt des Speichers zugegriffen wird.[12]

2.2.2 Angriffsszenarien auf die Microsoft Windows Hashfunktionen

Die drei Berechnungsschritte des LM-Hashs ermöglichen neue Angriffsformen: Es wird nicht wie oben die Hashfunktion angegriffen, sondern es werden die sicherheitstechnisch kritischen Schritte der Implementierung ausgenutzt. Der erste Schritt macht die Verwendung von Groß– und Kleinbuchstaben nutzlos und erhöht somit die Wahrscheinlichkeit eines erfolgreichen Angriffs, bei dem alle möglichen Zeichen-Kombinationen als Passwort nacheinander eingegeben, anschließend gehasht und mit dem Hash des gesuchten Passworts verglichen werden, bis eine Übereinstimmung erfolgt (brute force attack), weil sich die Zeichenanzahl um die der Kleinbuchstaben reduziert. Der dritte Schritt reduziert die maximale Länge aller Kombinationen bei der brute force attack auf 7 Zeichen, da jeder Teil des LM-Hashs separat angegriffen und das endgültige Passwort aus beiden Teilen anschließend zusammengesetzt wird.

Obwohl der NTML-Hash in der Theorie mehr Sicherheit gewährleistet, ist dies in der Praxis nur relativ der Fall. Weil standardmäßig beide Hashfunktionen aktiviert sind, hebt der LM-Hash bei Passwörtern mit weniger als oder genau 14 Zeichen die Verbesserungen des NTLM-Hashs auf. Sobald ein Angriff auf den LM-Hash erfolgreich war und das Passwort in Großbuchstaben (und eventuell Zahlen bzw. Sonderzeichen) bekannt ist, müssen die Zeichen des Passwortes lediglich noch per brute force attack auf den NTML-Hash auf Groß– und Kleinschreibung überprüft werden. Dies dauert in der Praxis meist wenige Sekunden.

---

10 Vgl. source code des Programms Bkhive, Cuomo (2004).

11 Vgl. source code des Programms Bkhive, Cuomo (2004).

12 Vgl. source code des Programms PwDump4, bingle (2003);
S. Abbildung 1.

Aber auch wenn der LM-Hash deaktiviert ist, gibt es einige Methoden, den NTML-Hash zu brechen. Weil viele Passwörter weniger als 16 Byte groß sind, werden diese mit Nullen gefüllt. Deswegen kann eine reine brute force attack hier ziemlich effizient sein, sie benötigt unter Umständen nur wesentlich mehr Zeit. Auf Grund dessen, dass die Hashfunktionen nicht salted[13] sind, das heißt, dass den Passwörtern vor dem Hashen keine Zufallswerte hinzugefügt werden, sondern entweder (je nach Länge des Passworts) mit Nullen aufgefüllt, gekürzt oder mit den Originalzeichen gehasht werden, ist es theoretisch möglich, alle möglichen Kombinationen im Vornherein ansatzweise zu berechnen und in einer Datenstruktur zu speichern, so dass der Hash des gesuchten Passworts mit den Hashes in den so genannten rainbow tables verglichen werden kann und das dazugehörige Passwort bei erfolgreichem Vergleich ausgegeben wird.

2.2.3 Rainbow tables

Der Einsatz von rainbow tables ist eine weitere effiziente Methode, entwickelt von Philippe Oechslin, um Hashes zu brechen. Rainbow tables sind im Voraus berechnete Hash-Tabellen. Zwei verschiedene Algorithmen sind für diese Datenstruktur bedeutsam: Zum einen der für die Generierung dieser, zum anderen der zum Finden des Klartextes (lookup) mit Hilfe dieser.

Der erste Algorithmus arbeitet wie folgt: Ein möglicher Klartext[14] wird durch die Hashfunktion (zum Beispiel die LM-Hashfunktion) zu einem Hashwert transformiert. Anschließend wird dieser durch eine Reduktionsfunktion zu einem neuen Klartext mit der gleichen Länge des Ursprünglichen abgebildet. Dieser Schritt wird n-mal wiederholt. Man nennt dies eine Kette (chain). Von jeder Kette wird der Anfangs- und Endwert gespeichert. Auch dies wird n-mal wiederholt, so dass es n Ketten gibt (die Länge aller Ketten ist konstant). Jeder einzelne Schritt der Ketten hat eine eigenständige Reduktionsfunktion, wobei jede Reduktionsfunktion bei dem gleichen Schritt aller n Ketten gleich ist. Auf diese Weise wird die Wahrscheinlichkeit einer Verflechtung der Ketten (Kollision) vermindert, weil die Reduktionsfunktionen nicht einheitlich sind und deshalb eine Übereinstimmung der Endwerte der Ketten, durch die Abbildung eines Hashs auf einen Klartext, wesentlich unwahrscheinlicher ist, weil eine Verflechtung theoretisch überall, praktisch aber nur bei den gleichen Reduktionsfunktionen, bei dem gleichen Schritt der Ketten, auftreten kann.

Bei der Findung des Klartextes zu dem gegebenen Hashwert passiert Nachstehendes: Der gegebene Hash wird mit der letzten Reduktionsfunktion auf einem Klartext abgebildet, welcher mit den Endwerten aller Ketten verglichen wird. Wenn keine Übereinstimmung erfolgt, wird der gegebene Hash, um eine Gleichheit mit dem vorletzten Klartext der Ketten zu überprüfen, mit der vorletzten Reduktionsfunktion reduziert, gehasht, mit der letzten Reduktionsfunktion auf den Endwert abgebildet und wieder mit den Endwerten aller Ketten verglichen. Dies wird so lange wiederholt, bis die Kette am Anfang angelangt ist oder bis eine Gleichheit gefunden wird. Beim ersten Fall ist ein erfolgreicher lookup fehlgeschlagen und wird beendet, beim zweiten Fall wird die Kette, bei der die Übereinstimmung mit dem Endwert gefunden wurde, mit Hilfe des Startwerts dieser bis zu der letzten von dem gegebenen Hash verwendeten Reduktionsfunktion neu aufgebaut (zur Laufzeit neu berechnet). Der vorhergehende Klartext ist der Gesuchte.[15]

13 Vgl. Peikari/Chuvakin (2004), S. 282.

14 „möglich“ bedeutet, dass der String nur die Zeichen und die Länge enthält, welche im Vornherein definiert wurden (z.B. nur Kleinbuchstaben, maximal sechs Zeichen), um die Treffgenauigkeit zu erhöhen und den Zeitaufwand zu minimieren. Jeder folgende Klartext erfüllt diese Kriterien.

15 Vgl. Bleikertz/Bugiel (2007), S. 23-26; für eine ausführliche Erläuterung vgl. Oechslin (2003).

Ophcrack ist ein von Philippe Oechslin geschriebenes Programm, welches Hashwerte mit Hilfe von rainbow tables bricht. Die *Ophcrack LiveCD* basiert auf der Linux-Distribution SLAX6 und beinhaltet sowohl Ophcrack als auch alpha-numerisch vorberechnete rainbow tables. Bootet man von der CD, startet SLAX6 automatisch Ophcrack. Letzteres liest die Hashes ähnlich wie PwDump4 aus und extrahiert sie, liest die rainbow tables ein und startet mit diesen den Angriff auf die LM-Hashes. Nach kurzer Zeit sind die alpha-numerischen Klartexte gefunden und es wird per brute force attack die Art der Buchstabenschreibung überprüft, damit eine Übereinstimmung mit dem gegebenen NTLM-Hash besteht.[16]

2.3 Schutzmaßnahmen

Bei einer Deaktivierung der Verwendung des standardmäßig aktivierten LM-Hashs wird eine Schwäche der Windows NT 5-Systeme abgeschaltet. Durch Verwendung starker Passwörter, zum Beispiel Passwörtern, welche mindestens aus 15 Zeichen, Groß- und Kleinbuchstaben, Zahlen und Sonderzeichen bestehen, wird ein Bruch des NTML-Hashs erschwert. Viele Linux-Distributionen wie zum Beispiel Debian haben mehrere Hashfunktionen zur Speicherung des Passworts implementiert, so dass der User frei wählen kann, welche er bevorzugt, und fügen standardmäßig einen salt hinzu, so dass brute force attacks erschwert und rainbow table attacks verhindert werden.

Eine Vollverschlüsselung der Betriebssysteme, zum Beispiel durch AES[17], verhindert jegliche Zugriffsmöglichkeiten auf das System außerhalb der Laufzeit.

16 Vgl. source code des Programms Ophcrack, Oechslin (2007);
S. Abbildung 2; S. Abbildung 3.

17 Vgl. Federal Information Processing Standards Publication 197 (fips-197).

3. LAN-interne Angriffsszenarien auf Microsoft Windows NT 5-Systeme

3.1 Das OSI-Referenzmodell

Das OSI-Referenzmodell (Open Systems Interconnection Reference Model) ist ein standardisiertes Schichtenmodell, welches als Designgrundlage für Kommunikationsprotokolle dient. Es besteht aus sieben aufeinander aufbauenden Schichten (layers), wobei jede Ebene eine genaue Funktion hat.

7. Application layer

Die Anwendungsschicht stellt Dienste und Anwendungen (zum Beispiel HTTP, HTTPS, FTP) zum Informationsaustausch bereit. Erhält sie die Daten von den unteren Schichten, leitet sie diese an eine Anwendung, über einen Socket, einem Ende einer Kommunikationsschnittstelle, weiter, erhält sie Daten von der Anwendung über den Socket, leitet sie die Daten an die untere Schicht weiter.[18]

6. Presentation layer

Die Darstellungsschicht gewährleistet durch Transformation der systemabhängigen Darstellung der Daten (zum Beispiel ASCII) der Anwendungsschicht des sendenden Systems in eine systemunabhängige Form, dass die Anwendungsschicht des empfangenden Systems die Daten lesen kann. Sie ermöglicht auch eine Kompression und Verschlüsselung der Daten.[19]

5. Session layer

Die Aufgabe der Sitzungsschicht ist es, die Kommunikation sowie den organisierten und synchronisierten Datenaustausch zwischen zwei Systemen zu ermöglichen. Durch check points wird die Wiederaufnahme einer zusammengebrochenen Sitzung an der Stelle des Abbruchs ermöglicht.[20]

4. Transport layer

Die Transportschicht ist für den Verbindungsaufbau und die Fehlersicherung und -behebung zuständig. Sie stellt den anwendungsorientierten Ebenen (5-7) einen einheitlichen Zugriff, eine Ende-zu-Ende-Verbindung zur Verfügung, so dass diese physische Eigenschaften oder Routing nicht berücksichtigen müssen.[21]

---

18 Vgl. ISO standard 7498-1:1994, 7.1.
19 Vgl. ISO standard 7498-1:1994, 7.2.
20 Vgl. ISO standard 7498-1:1994, 7.3.
21 Vgl. ISO standard 7498-1:1994, 7.4.

3. Network layer

Die Vermittlungsschicht ist für die logische Adressierung mit Quell- und Zieladressen (zum Beispiel IP-Adressen) zuständig. Des Weiteren erstellt und aktualisiert sie Routingtabellen (Tabellen mit Netzwerkknoten, welche den Datenverkehr weiterleiten, falls keine direkte Kommunikation zwischen Absender und Ziel möglich ist) und sorgt für die Flusskontrolle, das heißt, dass sichergestellt wird, dass die Datenübertragung ohne Verluste erfolgen kann.[22]

2. Data Link layer

Die Sicherungsschicht hat die Aufgabe, die physikalischen Adressen zuzuordnen, Datenübertragungsfehler zu erkennen und zu beheben sowie den Datenfluss zu kontrollieren.[23]

1. Physical layer

Die Bitübertragungsschicht arbeitet auf der Hardware-Ebene. Sie definiert die elektronische, mechanische und funktionale Schnittstelle zur physischen Verbindung, dem Übertragungsmedium. Erhält die erste Ebene Bitströme von der Sicherungsschicht, leitet sie die Daten zu einer anderen physischen Schnittstelle weiter, erhält sie die Daten von einem anderen physischen Medium, übergibt sie die Daten an die obere Schicht.[24]

Beim Versenden und Empfangen eines Datenpakets passiert jedes Paket alle sieben Schichten des OSI-Referenzmodells. Sendet das System ein Paket an einen Zielhost, wird das Paket von einem Programm auf einem bestimmten Port über einen Socket an die Anwendungsschicht übergeben. Diese leitet das Paket weiter, bis es an der ersten Schicht angekommen ist. Jede Schicht wird ineinander eingebettet, so dass das Paket bei jeder Schicht nach unten hin größer wird. Von der Bitübertragungsschicht aus wird es (eventuell über mehrere Knoten) zum Ziel geleitet. Beim Zielhost angekommen wird es durch die erste Schicht nach oben gesendet – nach oben hin wird das Paket kleiner - bis es in der Anwendungsschicht ankommt, welche das Datenpaket dann über den Socket durch den Zielport zur Anwendung übergibt.[25]

---

22 Vgl. ISO standard 7498-1:1994, 7.5.
23 Vgl. ISO standard 7498-1:1994, 7.6.
24 Vgl. ISO standard 7498-1:1994, 7.7.
25 S. Abbildung 4.

3.2 Die MAC-Adresse

3.2.1 Aufbau der MAC-Adresse

Eine MAC (Media Access Control) -Adresse ist eine (theoretisch) einmalige eindeutige Hardware-Adresse eines Netzwerkwerkadapters und dient zur Identifizierung. Bei manchen Netzwerkkarten (NIC - „Network Interface Card“) ist diese zum Beispiel auf dem EEPROM (Electrically Erasable Programmable Read Only Memory) gespeichert. Die MAC-Adresse fungiert zur Adressierung der Netzwerkadapter auf der Sicherungsschicht, damit an diese im Netzwerk zum Beispiel über Ethernet Pakete gesendet und empfangen werden können. Die MAC-Adresse ist ein 48 Bit (sechs Byte) großer Integer, welche im Detail wie folgt aufgebaut ist:

Die 48 Bit sind in zwei 24 Bit Blöcke geteilt. Der erste Block besteht aus der von der IEEE einmalig vergebenen Herstellerkennung[26], der zweite aus der vom Hersteller vergebenen Seriennummer.[27]

3.2.2 MAC-Spoofing

Softwareseitig lässt sich die Quell-Mac-Adresse bei Windows NT 5-Systemen beliebig verändern. Der Treiber des Herstellers einer Netzwerkkarte liest zur Laufzeit Windows die MAC-Adresse der Netzwerkkarte auf der Sicherungsschicht aus und übergibt sie der Network Driver Interface Specification (NDIS) von Windows[28], welche die MAC-Adresse in der Registry speichert. Die Programme changemac-win von Robbe De Keyzer[29] und MACAddressChanger von Nishant Sivakumar[30] ändern den Registry-Wert der MAC-Adresse und starten den Netzwerkadapter neu. Dies kann man auch ohne die Hilfe eines Programms machen. Beim Versenden eines Datenpakets wird diesem nun die gefälschte MAC-Adresse als Quell-MAC-Adresse eingetragen.

3.2.3 Möglichkeiten des MAC-Spoofing

Eine veränderte MAC-Adresse schafft größere Anonymität innerhalb eines Netzwerks, weil die Eindeutigkeit der originalen MAC-Adresse (temporär) nicht mehr vorhanden ist und eine Identifizierung des Netzwerkadapters und somit eines eventuellen Angreifers erschwert – wenn nicht sogar verhindert - wird. Außerdem ist es möglich, wenn man eine im LAN bereits vorhandene MAC-Adresse eines Clients annimmt, DHCP[31] im Netzwerk aktiviert ist und man deshalb die gleiche IP-Adresse des Clients zugewiesen bekommt, man also die Identität auf der zweiten Ebene (durch die MAC-Adresse) und auf der dritten Ebene (durch die IP-Adresse) angenommen hat, während der Client im Netzwerk aktiv ist, dass die Antwortpakete, die der Client erwartet, von einem Angreifer abgefangen werden,

---

26 Vgl. IEEE Registration Authority (2006).
27 Vgl. Wikipedia (2008b); kingpin (1998).
28 Vgl. Microsoft (2001);
S. Abbildung 5.
29 Vgl. source code des Programms changemac-win, De Keyzer (2005).
30 Vgl. source code des Programms MACAddressChanger, Sivakumar (2005);
S. Abbildung 6.
31 Vgl. RFC 2131.

sofern seine Verbindungszeit zum Gateway geringer ist als die des Clients. Damit dies der Fall ist, kann ein Angreifer zum Beispiel durch das Senden großer Datenpakete an den Client dessen Verbindung überlasten (DoS (denial-of-service) attack). Dieser muss nur sicherstellen, dass die Pakete an den Client und nicht an sich selbst geschickt werden.

Theoretisch kann er eine bestehende Sitzung (session) übernehmen, ohne sich zu authentifizieren (session hijacking).[32] Da dieser Angriff etwas komplizierter und umfangreicher ist, wird er hier nicht weiter ausgeführt. Die letzten Angriffe sind nur in Verbindung mit IP-Spoofing möglich.

3.3 Internet Protocol Version 4

3.3.1 Aufbau des IPv4

Das Internet Protocol Version 4 ist die vierte Version des Internetprotokolls. Es arbeitet auf der Vermittlungsschicht und ist unter anderem für die logische Adressierung der Endgeräte durch 32 Bit große IP-Adressen und für das Routing zuständig. Das Internetprotokoll versieht jedes Datenpaket, welches über dieses versendet werden soll (zum Beispiel TCP[33]-Pakete aus der Transportschicht) mit einem IP-Header. Dieser enthält unter anderem die Quell- und Zieladresse des Pakets und einer Prüfsumme (Header Checksum), welche unter anderem für die Fehlererkennung der Paketübertragung benötigt wird.[34]

3.3.2 IP-Spoofing

Winsock[35] ist eine API (application programming interface) für Programmierer, welche es ermöglicht, auf den raw socket zuzugreifen. Ein raw socket ist eine spezielle Art des Sockets, welcher es ermöglicht, Pakete auf der Transport- und der Vermittlungsschicht zu verändern bzw. zu erzeugen.[36] Deshalb kann man die Quell-IP-Adresse beliebig verändern. Weil nach der Veränderung die Header Checksum des IP-Headers nicht mehr übereinstimmt, muss dieser neu berechnet und in jedem gefälschten IP-Paket ausgetauscht werden.

3.3.3 Möglichkeiten des IP-Spoofing

IP-Spoofing schafft wie auch schon MAC-Spoofing Anonymität im Netzwerk, ermöglicht bei einer geringeren Verbindungszeit zum Kommunikationspartner wie die des Clients (sofern eine bereits im Netzwerk aktive IP-Adresse dauerhaft angenommen, nicht nur beim Versenden gefälscht wird (DHCP)) das Abfangen von Paketen und session hijacking. Weil alle Antwortpakete immer an die gefälschte Quell-IP-Adresse gesendet werden, erhält der Angreifer in der Regel keine Antworten. Ein Angreifer kann dies für eine distributed

---

32 Vgl. lkm (2007).
33 Vgl. RFC 793; RFC 1323.
34 Für eine detailliertere Beschreibung des Internetprotokolls und den Aufbau eines IP-Headers vgl. RFC 791.
35 Vgl. Microsoft (2008).
36 Vgl. Foster/Proce (2005), S. 312.

reflected denial-of-service attack (DRDoS attack), eine Überlastung eines Hosts durch den Empfang vieler Datenpakete von unterschiedlichen Rechnern ohne eine direkte Adressierung der Pakete durch den Angreifer an das Opfer, welche zu einem Absturz führen kann, nutzen, in dem er beispielsweise in einem großen LAN an mehrere Empfänger Anfragen mit der gefälschten IP-Adresse seines Opfers schickt, so dass dieses etliche unerwartete Antworten erhält, mit denen es nichts anfangen kann, und eventuell wegen Überlastung abstürzt. Bei einem solchen Angriff ist der Täter in der Regel schwer bis gar nicht zu identifizieren, weil er mit einer gefälschten IP-Adresse Pakete verschickte.

Es ist trotzdem möglich, dass der Angreifer beim Senden von Datenpaketen mit gefälschter Quell-IP-Adresse in einem LAN Antwortpakete erhält. Dazu muss er eine nicht vorhandene IP-Adresse benutzen (eine Vorhandene ist auch möglich, dazu benötigt er aber eine geringere Verbindungszeit als das Opfer zum Gateway) , welche sich im selben Subnetz befindet. Er schickt einen ARP request, welches die IP-Adresse des Zielsystems beinhaltet, an die MAC-Adresse des Broadcasts, FF:FF:FF:FF:FF:FF, um die MAC-Adresse des Gateways, zum Beispiel in Form eines Routers, zu erhalten. Alle Netzwerkteilnehmer erhalten diese Anfrage, und ein Empfänger, welcher die bei der Anfrage übermittelte Ziel-IP-Adresse hat (hier: das Gateway), antwortet dem Angreifer mit einer ARP-Antwort, welche die eigene MAC-Adresse erhält. Weil das Netzwerkprotokoll ARP jeder MAC-Adresse eine IP-Adresse zuordnet, in diesem Fall der originalen MAC-Adresse des Angreifers die gefälschte IP-Adresse, und der Versand der Pakete an die MAC-Adresse erfolgt (vgl. OSI-Modell), erhält der Angreifer Antwortpakete.

3.4 Address Resolution Protocol

3.4.1 Aufbau des ARP

Das Address Resolution Protocol ordnet Netzwerkadressen 48 Bit große Hardwareadressen zu, damit IP-Pakete (3. OSI-Schicht) in Ethernet-Pakete (2. OSI-Schicht) transformiert und an die Ziel-MAC-Adresse gesendet werden. Somit arbeitet ARP zwischen der Vermittlungs- und der Sicherungsschicht. Diese Zuordnungen werden aus Effizienz für eine gewisse Zeit in einer Tabelle, dem ARP-Cache, gespeichert. Ist zu einer bestimmten Netzwerkadresse keine Hardwareadresse im Speicher, wird diese erfragt.[37]

Die Kommunikation zwischen der dritten und der zweiten Schicht funktioniert wie folgt: Ein Datenpaket, welches auf der dritten Schicht angelangt ist, soll zu einem anderen Netzwerkpartner gesendet werden. Dafür muss eine Umwandlung in ein Ethernet-Paket, beispielsweise durch den Netzwerk- bzw. Hardwaretreiber, stattfinden. Dieser beauftragt das Address Resolution Protocol, ihm die zu der Ziel-Netzwerkadresse zugehörige Hardwareadresse zu geben. Letzteres vergleicht das Gegebene mit den Speicherinhalten und gibt, falls der Vergleich erfolgreich war, den dazugehörigen Wert aus. Der Treiber transformiert dann das Paket, übergibt es an die Hardware und verschickt es an die Hardwareadresse. Ist der Vergleich nicht erfolgreich, informiert das Protokoll seinen Auftraggeber, dass es das Paket verwirft, weil es davon ausgeht, dass eine höhere Schicht das Paket erneut sendet (wie zum Beispiel das Protokoll TCP (4. OSI-Schicht), welches Übertragungsfehler erkennt), generiert ein ARP-Paket, welches im Header neben Informationen wie dem Hardwareadresstyp, für den jetzt Ethernet eingesetzt wird, und dem Protokolladresstyp, für den jetzt IP eingesetzt wird, die Quell-MAC-Adresse, für welche jetzt die eigene (eventuell gefälschte) MAC-Adresse eingesetzt wird, die Quell-IP-

37 Vgl. RFC 826.

Adresse, für welche jetzt die eigene (eventuell gefälschte) IP-Adresse eingesetzt wird, die Ziel-MAC-Adresse, für welche jetzt die MAC-Adresse des Broadcasts eingesetzt wird (damit alle Netzwerkteilnehmer die Anfrage erhalten), und die Ziel-IP-Adresse, für welche die IP-Adresse des Hosts, dessen MAC-Adresse gesucht wird, eingesetzt wird, enthält (ARP request), und übergibt es dem Netzwerkmodul (zum Beispiel der Netzwerkkarte), welche es dann an das Broadcast sendet.[38] Die Netzwerkmodule jedes Netzwerkteilnehmers erhalten diese Anfrage und leiten sie zu dem Address Resolution Protocol weiter, welches die Eigenschaften (ungefähr, weil nicht alle Optionen des ARP-Paket-Headers aus Gründen der fehlenden Notwendigkeit für die folgenden Ausführungen genannt wurden bzw. werden) auf folgende Weise überprüft:

Zuerst wird überprüft, ob der Protokolladresstyp und die Sender-IP-Adresse im ARP-Cache bereits vorhanden sind. Falls dies zutrifft, wird die Sender-MAC-Adresse aktualisiert, falls nicht, wird die Aktualisierung übergangen. Anschließend wird die Ziel-IP-Adresse mit der eigenen verglichen. Stimmt diese überein, werden zuerst, falls der Protokolladresstyp und die Sender-IP-Adresse noch nicht im ARP-Cache vorhanden sind, diese mit der Sender-MAC-Adresse in diesem eingetragen, ansonsten wird direkt kontrolliert, ob der Inhalt der Status-Variable *opcode* des ARP-Pakets *request* ist. Wenn dies der Fall ist, werden die Sender-MAC-Adresse und die Sender-Ziel-Adresse durch die eigenen, die Quell-MAC-Adresse und die Quell-IP-Adresse durch die des Senders, und der Wert der Status-Variable *opcode* auf *reply* gesetzt, das Paket an das Netzwerkmodul übergeben und an die Ziel-MAC-Adresse gesendet. Der Empfänger, welcher den ARP request sandte, durchläuft nach Weiterleitung des Paketes an das Address Resolution Protocol den gleichen Algorithmus. Trifft irgendeine Bedingung nicht zu, wird das Paket verworfen.[39]

Man unterscheidet zwischen dynamischen und statischen Einträgen im ARP-Cache. Dynamische Einträge werden automatisch durch ARP-Pakete hinzugefügt bzw. aktualisiert und bei längerer Nicht-Verwendung gelöscht. Statische Einträge werden manuell verwaltet (hinzugefügt, aktualisiert, gelöscht). Microsoft Windows NT 5-Systeme löschen inaktive dynamische Einträge nach zwei, aktive dynamische Einträge nach zehn Minuten.[40]

In der Praxis sind dynamische Einträge weiter verbreitet, weil sie keinen administrativen Aufwand erfordern. Statische Einträge hingegen müssen auf jedem Rechner für jeden Rechner im Netzwerk angelegt werden. Mit der Anzahl der Computer im Netzwerk steigt die Anzahl der manuell einzugebenden MAC-Adressen mit den dazugehörigen IP-Adressen quadratisch. Des Weiteren ergeben eigene Forschungen, dass Windows NT 5-Systeme auch bei statischen Einträgen nach manipulierten ARP-Antworten temporär diese überschreiben und somit anfällig für ARP-Spoofing sind.

3.4.2 ARP-Spoofing

Die bereits beschriebenen Algorithmen weisen sicherheitstechnisch eine Schwäche auf: Wegen einer fehlenden Authentifizierung des Absenders und einer Aktualisierung der

38 Vgl. RFC 826, Packet Generation.
39 Vgl. RFC 826, Packet Reception;
S. Abbildung 7;
S. Abbildung 8.
40 Vgl. Microsoft (2005b).

MAC-Adresse in der Tabelle bei einem ARP request bzw. einem ARP response ist es möglich, mit Hilfe des Zugriffs auf den raw socket ARP-Pakete (mit falschen Absenderadressen) zu erzeugen oder andere ARP-Pakete zu verändern (manipulieren). ARP-Spoofing funktioniert auf zwei Arten:

Client A hat in seinem ARP-Cache die MAC- und IP-Adresse des Gateways B, und B hat diese in seinem von A. Der Angreifer C sendet an jede IP im Subnet einen ARP request mit der MAC-Adresse des Broadcasts als Ziel-MAC-Adresse, um zu erfahren, wer alles im Netzwerk erreichbar ist. Jeder verfügbare Teilnehmer, wie A und B, empfängt das Paket und geht den oben beschriebenen Algorithmus durch. Am Ende haben A und B C im ARP-Cache und C hat durch deren Antwortpakete ihre IP- mit den dazugehörigen MAC-Adressen.

Jetzt schickt C einen unaufgeforderten ARP response (gratuitous ARP), bei dem er für die Quell-IP-Adresse die IP-Adresse von B einsetzt, an A. A überschreibt die zur IP-Adresse von B gehörende MAC-Adresse mit der Quell-MAC-Adresse, der Adresse von C, im ARP-Cache, vergleicht den opcode und verwirft das Paket. Anschließend schickt C ein gratuitous ARP Paket mit der IP-Adresse von A als Quell-IP-Adresse an B, so dass B die MAC-Adresse von A mit der von C im ARP-Cache überschreibt und das Paket verwirft.

Bei der zweiten Methode sendet C, nach dem er alle IP- mit den dazugehörigen MAC-Adressen bekommen hat, einen ARP request mit der IP-Adresse von B als Quell-IP-Adresse an A, so dass A den Wert im ARP-Cache wieder überschreibt, aber diesmal der opcode übereinstimmt und ein ARP response, welcher als Ziel-IP-Adresse die von B hat, an die MAC-Adresse von C geschickt wird, so dass das Antwortpaket C erhält (vgl. OSI-Modell). Das gleiche schickt C an B, nur dass dort die Quell-IP-Adresse die von A ist und der ARP response, welcher als Quell-IP-Adresse die von A hat und an die MAC-Adresse von C übermittelt wird.

In beiden Fällen wird die Kommunikation über den Angreifer umgeleitet. Diese Attacke bezeichnet man als man-in-the-middle attack (MITM). Damit der Datenverkehr zwischen A und B weiterhin Zustande kommt, muss der Angreifer die Datenpakete an das ursprüngliche Ziel weiterleiten. Dazu ändert er die Ziel-MAC-Adresse und trägt eventuell, um nicht aufzufallen, bei der Quell-IP- und -MAC-Adresse die des ursprünglichen Absenders ein. Weil der ARP-Cache automatisch nach einiger Zeit gelöscht wird, wiederholt der Angreifer das Senden manipulierter ARP-Pakete in einem geringen Intervall.

Durch die Verwendung einer gefälschten IP- und MAC-Adresse wird die Auffindung eines Angreifers immens erschwert.

3.4.3 Möglichkeiten des ARP-Spoofing

Weil sämtlicher Datenverkehr über den Angreifer umgeleitet wird, stehen ihm viele Möglichkeiten offen. Er kann beispielsweise alles, was im Klartext (plain text) verschlüsselt wird, mitschneiden (sniffen). Dies ermöglicht ihm unter anderem das Abfangen von Passwörtern, E-Mails[41] oder die Einsicht, welche Internetseiten das bzw. die Opfer aufrufen. Verschlüsselte Verbindungen kann er ebenfalls mitschneiden und versuchen, diese zu entschlüsseln (zum Beispiel durch Schwächen im Verschlüsselungsalgorithmus oder durch das Brechen schwacher Passwörter per brute force attack). Bei verschlüsselten Verbindungen, bei denen eine Authentifizierung durch

41 S. Abbildung 9.

Zertifikate stattfindet, wie zum Beispiel SSH[42] oder manche Implementationen von TSL, lassen sich die Zertifikate durch Client- und Server-Implementierungen der Protokolle fälschen. Das Opfer wird gewarnt, dass sich die Prüfsumme des Zertifikats geändert hat. Dies ist sowohl nach einer Manipulation als auch nach einem normalen Software-Update der Fall. Es liegt nun am Opfer, ob es die Verbindung abbricht oder das gefälschte Zertifikat annimmt. Nimmt es es an, kann der Angreifer die Zugangsdaten abfangen, mit seiner Client-Implementierung eine Verbindung zu dem Ziel-Server mit den abgefangenen Daten aufbauen und alles an das Opfer weiterleiten. So liest er sowohl unverschlüsselt die scheinbar verschlüsselte Kommunikation mit und ist im Besitz der für den Login benötigten Daten. Des Weiteren ist der Angreifer in der Lage, Inhalte zu manipulieren. Er kann beispielsweise Nachrichteninhalte verändern oder einen Proxy aufsetzen, welcher Webinhalte filtert oder verändert. Er kann unter anderem Phishing, eine Form des Social Engineering[43], der Manipulation des Menschen, ausüben, das heißt, er kann durch eine Filtereinstellung bestimmte Internetseiten, zum Beispiel Online-Banking-Sites auf eine manipulierte Webseite, welche ähnlich wie die Originale aussieht, umleiten und die Login-Daten oder sogar Transaktionsnummern abfangen. Des Weiteren kann man ein spezielles Opfer aus einer bestehenden session ausklinken, in dem man an diesen keine Pakete weiterleitet. Diese session kann dann der Angreifer übernehmen (session hijacking). Durch das Umleiten sämtlichen Datenstroms aller Netzwerkteilnehmer auf einen einzigen Client kann dieser durch eine DoS attack überlastet werden. ARP-Spoofing kann die unterschiedlichsten Folgen haben, die Angriffsmöglichkeiten sind theoretisch unbegrenzt und in ihrer Wirkung nicht vorhersehbar.

3.4.4 Funktionsweise eines Sniffer

Ein Sniffer ermöglicht das Mitschneiden und die Analyse des Netzwerkverkehrs, welchen das Netzwerkinterface, zum Beispiel die Netzwerkkarte, empfängt. Normalerweise wird eine Netzwerkkarte im non-promiscous mode betrieben, das heißt, dass sie nur die Pakete annimmt, welche an sie adressiert sind. Durch das Einschalten des promiscous mode nimmt sie auch Pakete an, die nicht an sie adressiert sind.[44] In Netzwerken, in welchen Switches zur Paketverteilung zum Einsatz kommen, empfängt die Netzwerkkarte einzig die Pakete, welche für sie bestimmt sind. Durch Angriffe wie MAC-, IP- und ARP-Spoofing wird dies geändert.

Ein Sniffer besteht aus drei Elementen: Einem Puffer (buffer), einem Filter und einem Treiber. Der Puffer dient zur Zwischenspeicherung der Datenpakete. Der Filter greift auf diese zu und analysiert sie. Der Treiber kommuniziert mit der Netzwerkkarte und übergibt die Pakete dem Puffer.

Als Programmbeispiel fungiert hier Wireshark.[45] Einen eigenen Treiber hat es nicht. Es nutzt die Programmbibliothek (API) WinPcap.[46] WinPcap ermöglicht unter anderem das Abfangen der Pakete von der Netzwerkkarte, aber auch, Daten selbst in das Netzwerk zu senden.[47] Es kommuniziert über NDIS direkt mit der Netzwerkkarte. Dies ermöglicht das Abfangen der Pakete im Rohformat auf der Sicherungsschicht, bevor das Betriebssystem auf die Datenpakete zugreift und sie verarbeitet (an die höheren Schichten transformiert

42 Vgl. RFC 4250-4256.
43 Vgl. Mitnick/Simon (2003).
44 Vgl. Rodewig (2006), S. 413.
45 Vgl. Combs (2008).
46 Vgl. Varenni/Degioanni/Risso/Bruno (2008).
47 Vgl. The WinPcap Team (2007), WinPcap user's manual.

übergibt).[48] Wireshark greift auf die Pakete im Rohformat über WinPcap zu und speichert sie auf der Festplatte.[49] Der Filter *Epan* analysiert die einzelnen Pakete, teilt sie auf, sortiert und transformiert sie. Er kann zum Beispiel ein einzelnes TCP-Paket darstellen und auf Wunsch den gesamten Verkehrsbaum des Protokolls mit einem bestimmten Kommunikationspartner anzeigen.[50]

3.5 Schutzmaßnahmen

Programme wie arpwatch[51] analysieren nach einer Lernzeit alle ARP-Anfragen und -Antworten am eigenen Netzwerkinterface und merken sich MAC-Adressen mit den dazugehörigen IP-Adressen. Ändert sich ein Adresspaar, meldet es dies durch einen Warnhinweis. Insofern ist diese Methode nur sinnvoll, wenn regelmäßig Log-Dateien gelesen werden. Ein Intrusion Detection System (IDS) wie Snort[52] arbeitet ähnlich wie arpwatch, nur dass es wesentlich mehr Aufgaben erfüllen kann. Ein IDS arbeitet nach festen Filter-Regeln und kann Angriffe unterschiedlichster Art nach bestimmten Mustern erkennen und gibt Warnungen aus. In Verbund mit richtig konfigurierten Firewalls kann so unter Umständen die Wahrscheinlichkeit eines Angriffs reduziert werden. Durch Verschlüsselung der Kommunikation, zum Beispiel die Authentifizierung per POP3[53] über TSL, die Verwendung von SSH-Tunneln oder VPNs (Virtual Private Networks) für die Kommunikation im LAN, zum Beispiel für die Dateiübertragung, die Verschlüsselung der Gesprächsinhalte zum Beispiel per OpenPGP[54], und die Verschlüsselung der Kommunikationsdaten nach Außen, zum Beispiel des HTTP[55]-Verkehrs über das TOR[56]-Netzwerk, bei dem die Kommunikation über mindestens drei unbekannte Rechner verschlüsselt übertragen wird, so dass keiner außer dem letzten Rechner, über den die Inhalte unverschlüsselt, sofern man kein verschlüsseltes Protokoll wie HTTPS[57] benutzt, in das Internet übertragen werden, keiner auf den Inhalt der Daten Zugriff hat bzw. in diese hineinsehen kann und durch die Schaltung mehrerer Rechner hintereinander eine Identifizierung des Benutzers bisher nur in der Theorie möglich ist, werden die Inhalte geschützt, so dass ein Angreifer die abgefangenen Daten nicht lesen kann. Eine effiziente Methode, innerhalb eines lokalen Netzwerkes IP- und ARP-Spoofing zu verhindern, ist der Einsatz von Ipsec[58], welches nach der Authentifizierung eine verschlüsselte Verbindung zur Gegenstelle auf der Vermittlungsschicht - im Vergleich zu den erwähnten Möglichkeiten, welche auf der Sitzungsschicht agieren - aufbaut, oder IPv6[59], welches Ipsec integriert hat und auf Grund einer anderen Architektur das ARP- und DHCP-Protokoll nicht mehr benötigt.

48 Vgl. The WinPcap Team (2007), WinPcap internals.
49 Vgl. Lamping (2007), 6.3. Capturing packets.
50 Vgl. Lamping (2007), 6.2. Overview.
51 Vgl. Lawrence Berkeley National Laboratory Network Research Group (2002).
52 Vgl. The Snort Team (2008).
53 Vgl. RFC 1939.
54 Vgl. RFC 4880.
55 Vgl. RFC 2616.
56 Vgl. The Tor Project (2008).
57 Vgl. RFC 2818.
58 Vgl. RFC 2401.
59 Vgl. RFC 2460.

4. Zusammenfassung

Alle demonstrierten lokalen Angriffsszenarien auf Microsoft Windows NT 5.0-, 5.1- und 5.2-Systeme zeigen, dass eine starke Passwortsicherheit nicht gegeben ist. Die implementierten Hashfunktionen sind nicht sicher, die Unsicherere ist aus Gründen der Abwärtskompatibilität standardmäßig aktiviert. Die Hashes werden nicht salted, wodurch ein Angriff durch die brute force attack erleichtert und durch rainbow tables ermöglicht wird.

Die Netzwerkprotokolle verfügen über keine Mechanismen, welche die Authentizität des Absenders gewährleisten. Deshalb ermöglichen Angriffsszenarien wie ARP-Spoofing eine Vielzahl von Angriffen, welche eine große Gefährdung für die Datensicherheit oder Ähnliches sein können. Durch Methoden wie MAC- und IP-Spoofing wird eine Identifizierung des Angreifers wesentlich erschwert.

Wie jede Art der Information kann man die vorgestellten Methoden dazu nutzen, um Schaden anzurichten, aber auch, um die Funktionsweise zu verstehen und daraufhin Präventionsmaßnahmen einzuführen.

Sicherheit ist kein Zustand, sondern ein Prozess. Die Realisierung der erwähnten Schutzmaßnahmen kann erfolgreiche Angriffe verringern, muss es aber nicht. Diese kann sehr kompliziert und aufwändig sein, dennoch umgangen werden. Entweder werden Fehler bei der Umsetzung der Präventionsmaßnahmen entdeckt oder es werden andere Angriffspunkte genutzt oder neue Fehler gefunden.

Jedes Betriebssystem hat Schwachstellen, jedes System ist und bleibt angreifbar. Laut Philippe Oechslin funktioniere der Angriff mit Ophcrack beispielsweise auch bei Windows NT 6 (Vista).[60]

---

60 Vgl. Oechslin (2007).

5. Literaturverzeichnis

Federal Information Processing Standard:

fips-197.

International Organization for Standardization:

ISO standard 7498-1:1994.

Internet:

Combs, Gerald (2008): Wireshark, http://www.wireshark.org, 27.02.2008, abgerufen am 27.02.2008.

Glass, Eric (2006): The NTLM Authentication Protocol and Security Support Provider, http://davenport.sourceforge.net/ntlm.html, 2006, abgerufen am 27.02.2008.

IEEE Registration Authority (2006): IEEE OUI and Company_id Assignments, http://standards.ieee.org/regauth/oui/index.shtml, 14.03.2006, abgerufen am 27.02.2008.

kingpin (1998): MAC Address Cloning, http://packetstormsecurity.org/docs/hack/mac_address_cloning.pdf, 25.12.1998, abgerufen am 27.02.2008.

Lamping, Ulf (2007): Wireshark Developer's Guide, 24493 for Wireshark 0.99.7, http://www.wireshark.org/docs/wsdg_html_chunked/, 2007, abgerufen am 27.02.2008.

Lawrence Berkeley National Laboratory Network Research Group (2002): LBNL's Network Research Group, http://www-nrg.ee.lbl.gov, 09.2002, abgerufen am 27.02.2008.

lkm (2007): Remote blind TCP/IP spoofing, Blind TCP/IP hijacking is still alive, in: Phrack-Magazine, 2007, Nr. 64, http://www.phrack.org/issues.html?issue=64&id=15&mode=txt, 27.05.2007, abgerufen am 27.02.2008.

Microsoft (2001): Network Driver Interface Specification, NDIS 5.0 Overview, http://www.microsoft.com/whdc/archive/ndis5.mspx, 04.12.2001, abgerufen am 27.02.2008.

Microsoft (2004a): Die Grundsatzdiskussion: Kennwort-Sätze oder Kennwörter, Problemstellung, http://www.microsoft.com/germany/technet/sicherheit/newsletter/kennwoerter.mspx, 17.09.2004, abgerufen am 27.02.2008.

Microsoft (2004b): Windows NT-Systemschlüssel erlaubt starke Verschlüsselung des SAM, http://support.microsoft.com/?scid=kb%3Bde%3B143475&x=22&y=6, 09.11.2004, abgerufen am 27.02.2008.

Microsoft (2005a): Windows XP Professional Resource Kit, Troubleshooting the Startup Process, http://technet.microsoft.com/en-us/library/bb457123.aspx, 03.11.2005, abgerufen am 27.11.2008.

Microsoft (2005b): Anzeigen des ARP-Caches (Address Resolution Protocol), http://technet2.microsoft.com/windowsserver/de/library/cae0e239-267b-45df-88a8-f6f1303830471031.mspx?mfr=true, 21.01.2005, abgerufen am 27.02.2008.

Microsoft (2006a): SO WIRD'S GEMACHT, Tipps und Tricks zur Registrierung (Registry) unter Windows XP Home Edition, Teil 1, http://support.microsoft.com/kb/822890/DE/, 04.07.2006, abgerufen am 27.02.2008.

Microsoft (2008): Windows Sockets 2, http://msdn2.microsoft.com/en-us/library/ms740673.aspx, 19.02.2008, abgerufen am 27.02.2008.

Oechslin, Philippe (2003): Making a Faster Cryptanalytic Time-MemoryTrade-Off, http://lasecwww.epfl.ch/

pub/lasec/doc/Oech03.pdf, 26.05.2003, abgerufen am 27.02.2008.

The Snort Team (2008): What is Snort?, http://www.snort.org/, 26.02.2008, abgerufen am 27.02.2008.

The Tor Project (2008): Tor: anonymity online, https://www.torproject.org, 27.02.2008, abgerufen am 27.02.2008.

The WinPcap Team (2007): WinPcap Documentation, 4.0.2, http://www.mirrorservice.org/sites/ftp.wiretapped.net/pub/security/packet-capture/winpcap/docs/docs_40_2/html/main.html, 2007, abgerufen am 27.02.2008.

Varenni, Gianluca; Degioanni, Loris; Risso, Fulvio; Bruno, John (2008): WinPcap, The Windows Packet Capture Library, http://www.mirrorservice.org/sites/ftp.wiretapped.net/pub/security/packet-capture/winpcap/default.htm, 31.01.2008, abgerufen am 27.02.2008.

Wikipedia (2008a): Geburtstagsparadoxon, http://de.wikipedia.org/wiki/Geburtstagsparadoxon, 01.01.2008, abgerufen am 27.02.2008.

Wikipedia (2008b): MAC-Adresse, Aufbau, Herstellerkennungen, 25.02.2008, abgerufen am 27.02.2008.

Wikipedia (2008c): OSI-Modell, Die 7 Schichten, http://de.wikipedia.org/wiki/OSI-Modell, 26.02.2008, abgerufen am 27.02.2008.

Printmedien:

Bleikertz Sören; Bugiel Sven (2007): Over the Rainbow, in: hakin9 – Abwehrmethoden, 2007, Nr. 5, S. 20-27.

Foster, James C.; Proce, Mike (2005): Sockets, Shellcode, Porting, & Coding, Reverse Engineering Exploits and Tool Coding for Security Professionals, 2005 Rockland, MA.

Mitnick, Kevin; Simon, William (2003): Die Kunst der Täuschung, Risikofaktor Mensch, 2003 Bonn.

Peikari, Cyrus; Chuvakin, Anton (2004): Kenne deinen Feind, Fortgeschrittene Sicherheitstechniken, Köln 2004.

Rodewig, Klaus M. (2006): Netzwerke mit Linux, Hochverfügbarkeit, Sicherheit und Perfomance, 2006 Köln.

Request for Comments:

RFC 791.
RFC 793.
RFC 826.
RFC 1323.
RFC 1939.
RFC 2131.
RFC 2401.
RFC 2460.
RFC 2616.
RFC 2818.
RFC 4250.
RFC 4251.
RFC 4252.
RFC 4253.
RFC 4254.
RFC 4255.
RFC 4256.
RFC 4880.

Quelltexte:

bingle (2003): PwDump4, http://www.mirrors.wiretapped.net/security/host-security/john/contrib/win32/pwdump/pwdump4.zip, 25.09.2003, abgerufen am 27.02.2008.

Cuomo, Nicola (2004): Bkhive, http://www.studenti.unina.it/~ncuomo/syskey/Bkhive.zip, 28.03.2004, abgerufen am 27.02.2008.

De Keyzer , Robbe (2005): changemac-win, http://packetstormsecurity.org/Win/changemac-win.c, 31.12.2005, abgerufen am 27.02.2008.

Oechslin, Philippe (2007): Ophcrack, http://ophcrack.sourceforge.net, 02.08.2007, abgerufen am 27.02.2008.

Sivakumar, Nishant (2005): MACAddressChanger, http://www.codeproject.com/KB/applications/MacIdChanger/MACAddressChanger_Src.zip, 25.05.2005, abgerufen am 27.02.2008.

.

6. Anhang

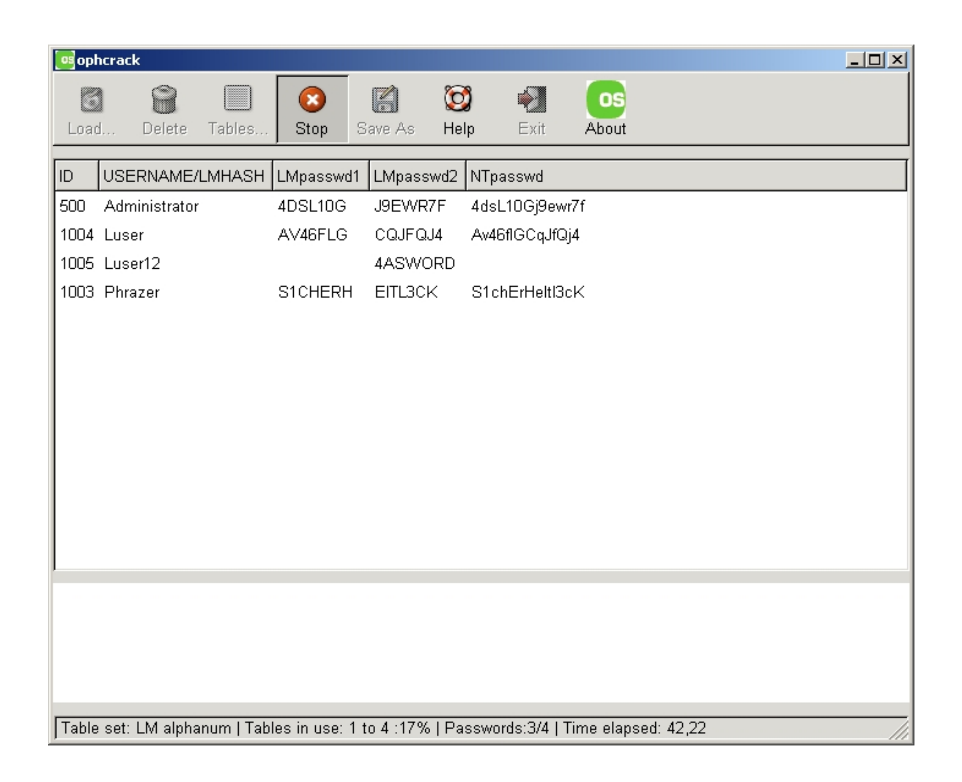
```
C:\Dokumente und Einstellungen\Phrazer\Desktop>pwdump4 127.0.0.1

PWDUMP4.02 dump winnt/2000 user/password hash remote or local for crack.
   by bingle@email.com.cn
This program is free software based on pwpump3 by Phil Staubs
under the GNU General Public License Version 2.

local path of \\127.0.0.1\ADMIN$ is: C:\WINDOWS
connect to 127.0.0.1 for result, plz wait...
SRV>Version: OS Ver 5.1, Service Pack 2, Workstation
Administrator:500:B56A60D330AEEE2932B17F5D197EB4DD:15C8BCED7CC3204F3F482D18A726B
D4B:::
Gast:501:AAD3B435B51404EEAAD3B435B51404EE:31D6CFE0D16AE931B73C59D7E0C089C0:::
Hilfeassistent:1000:B607B77A1845D5CC63AC8AACC4D0997F:82C1CD62B2ED658EE9B649FEEFB
06A7E:::
Luser:1004:47939F268F9A5AA0DEC90601799C00C2:6F87AEEE161185E1A131560B14AE00B6:::
Luser12:1005:6A8703DFD3DE2E7EF9A6E4AA6F5CF6E0:98061517F11B0A2E5CB21C830099BD01::
:
Phrazer:1003:4E1E3F575A29EF4DDA63114B02B1194A:3DF43C9780988A65892F4AB3269630ED::
:
SUPPORT_388945a0:1002:AAD3B435B51404EEAAD3B435B51404EE:0E5822A11E39C453EBF131C3D
C9F3A2A:::
LSA>Samr Enumerate 7 Users In Domain HOME-PC.
 All Completed.

C:\Dokumente und Einstellungen\Phrazer\Desktop>
```

Abbildung 1: Hashes auslesen mit PwDump4



ID	USERNAME/LMHASH	LMpasswd1	LMpasswd2	NTpasswd
500	Administrator	4DSL10G	J9EWR7F	4dsL10Gj9ewr7f
1004	Luser	AV46FLG	CQJFQJ4	Av46flGCqJfQj4
1005	Luser12		4ASWORD	
1003	Phrazer	S1CHERH	EITL3CK	S1chErHeltl3cK

Table set: LM alphanum | Tables in use: 1 to 4 :17% | Passwords:3/4 | Time elapsed: 42,22

Abbildung 2: Mit Ophcrack zur Hälfte gebrochener Hash

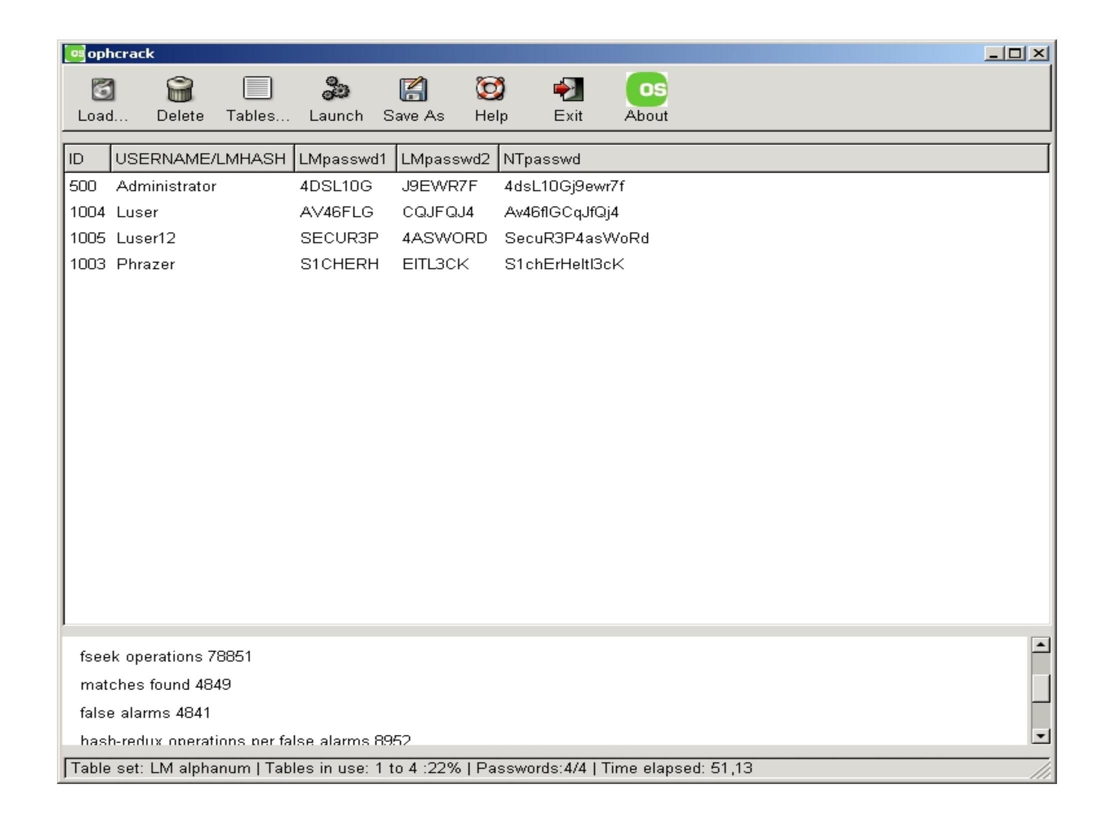


Abbildung 3: Mit Ophcrack in kurzer Zeit gebrochene Hashes

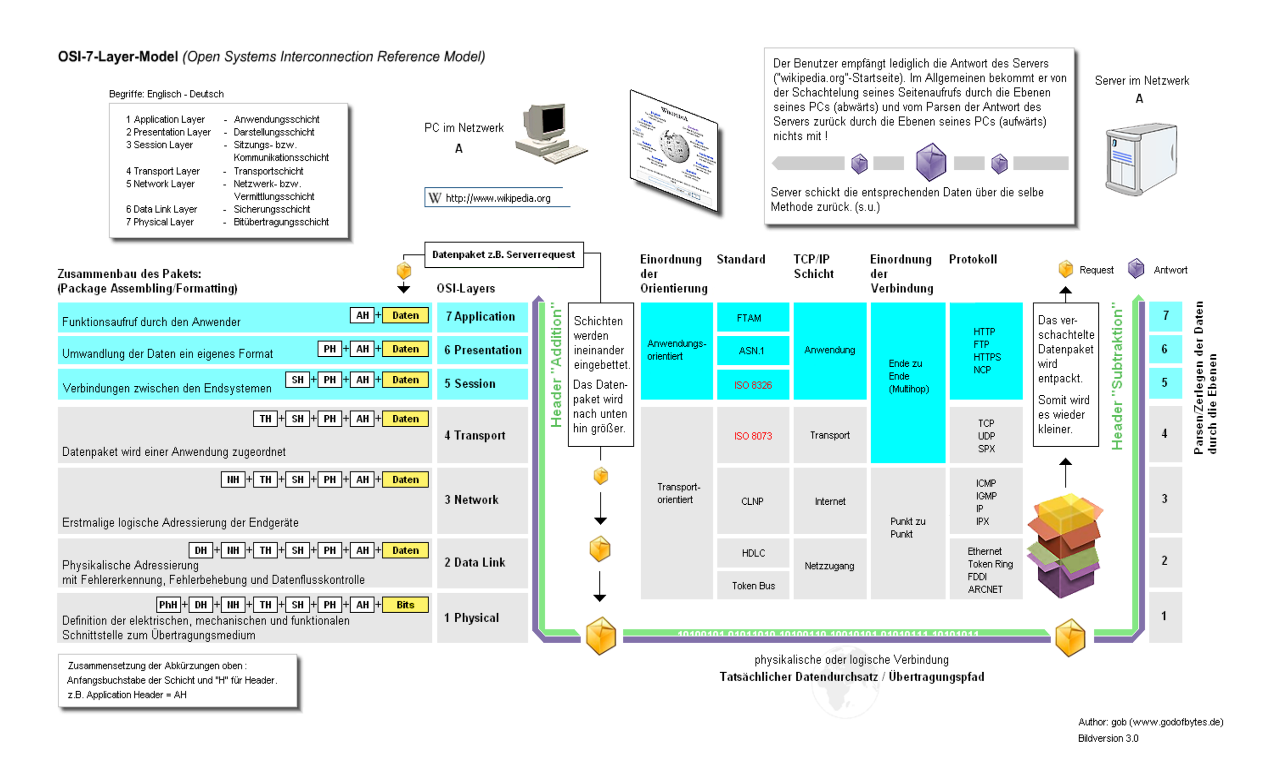


Abbildung 4: Das OSI-Schichtenmodell

Quelle: Wikipedia (2008c).

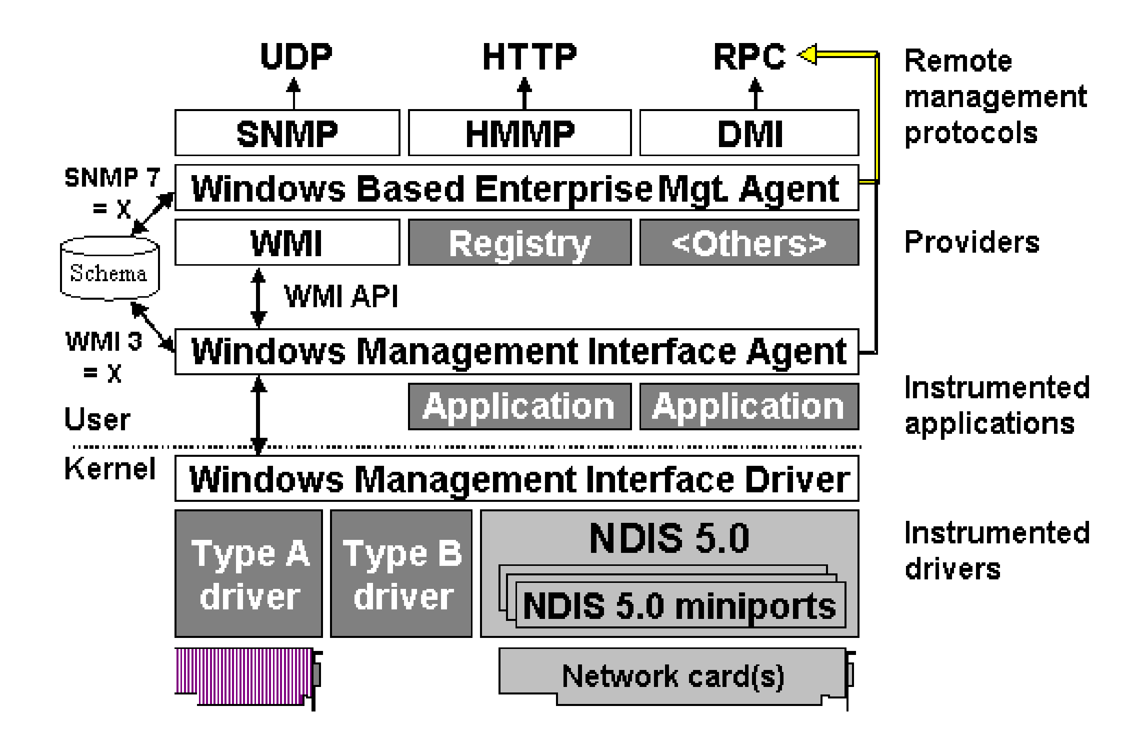


Abbildung 5: Kommunikation zwischen NDIS und der Netzwerkkarte

Quelle: Microsoft (2001).

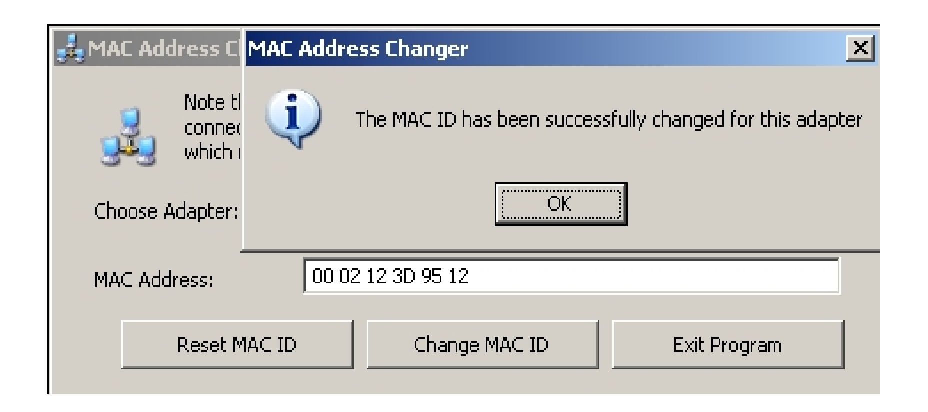


Abbildung 6: Fälschen einer MAC-Adresse

```
Frame 15 (42 bytes on wire, 42 bytes captured)
  Arrival Time: Feb 28, 2008 20:23:46.264250000
  [Time delta from previous captured frame: 0.015617000 seconds]
  [Time delta from previous displayed frame: 0.015617000 seconds]
  [Time since reference or first frame: 0.218720000 seconds]
  Frame Number: 15
  Frame Length: 42 bytes
  Capture Length: 42 bytes
  [Frame is marked: False]
  [Protocols in frame: eth:arp]
  [Coloring Rule Name: ARP]
  [Coloring Rule String: arp]
Ethernet II, Src: Sierraco_3d:95:12 (00:02:12:3d:95:12), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊟ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    Address: Broadcast (ff:ff:ff:ff:ff:ff)
    .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
    .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
⊟ Source: Sierraco_3d:95:12 (00:02:12:3d:95:12)
    Address: Sierraco_3d:95:12 (00:02:12:3d:95:12)
    .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
  Type: ARP (0x0806)
Address Resolution Protocol (request)
  Hardware type: Ethernet (0x0001)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (0x0001)
  Sender MAC address: Sierraco_3d:95:12 (00:02:12:3d:95:12)
  Sender IP address: 192.168.15.3 (192.168.15.3)
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.15.15 (192.168.15.15)
```

Abbildung 7: ARP request mit gefälschter MAC-Adresse

```
⊟ Frame 16 (60 bytes on wire, 60 bytes captured)
    Arrival Time: Feb 28, 2008 20:23:46.264589000
    [Time delta from previous captured frame: 0.000339000 seconds]
    [Time delta from previous displayed frame: 0.000339000 seconds]
    [Time since reference or first frame: 0.219059000 seconds]
    Frame Number: 16
    Frame Length: 60 bytes
    Capture Length: 60 bytes
    [Frame is marked: False]
    [Protocols in frame: eth:arp]
    [Coloring Rule Name: ARP]
    [Coloring Rule String: arp]
⊟ Ethernet II, Src: 192.168.15.15 (00:08:9f:0a:0a:0f), Dst: Sierraco_3d:95:12 (00:02:12:3d:95:12)
  ⊟ Destination: Sierraco_3d:95:12 (00:02:12:3d:95:12)
      Address: Sierraco_3d:95:12 (00:02:12:3d:95:12)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
  ⊟ Source: 192.168.15.15 (00:08:9f:0a:0a:0f)
      Address: 192.168.15.15 (00:08:9f:0a:0a:0f)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Type: ARP (0x0806)
    Trailer: 000000000000000000000000000000000000
⊟ Address Resolution Protocol (reply)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (0x0002)
    Sender MAC address: 192.168.15.15 (00:08:9f:0a:0a:0f)
    Sender IP address: 192.168.15.15 (192.168.15.15)
    Target MAC address: Sierraco_3d:95:12 (00:02:12:3d:95:12)
    Target IP address: 192.168.15.3 (192.168.15.3)
```

Abbildung 8: ARP response an die gefälschte MAC-Adresse

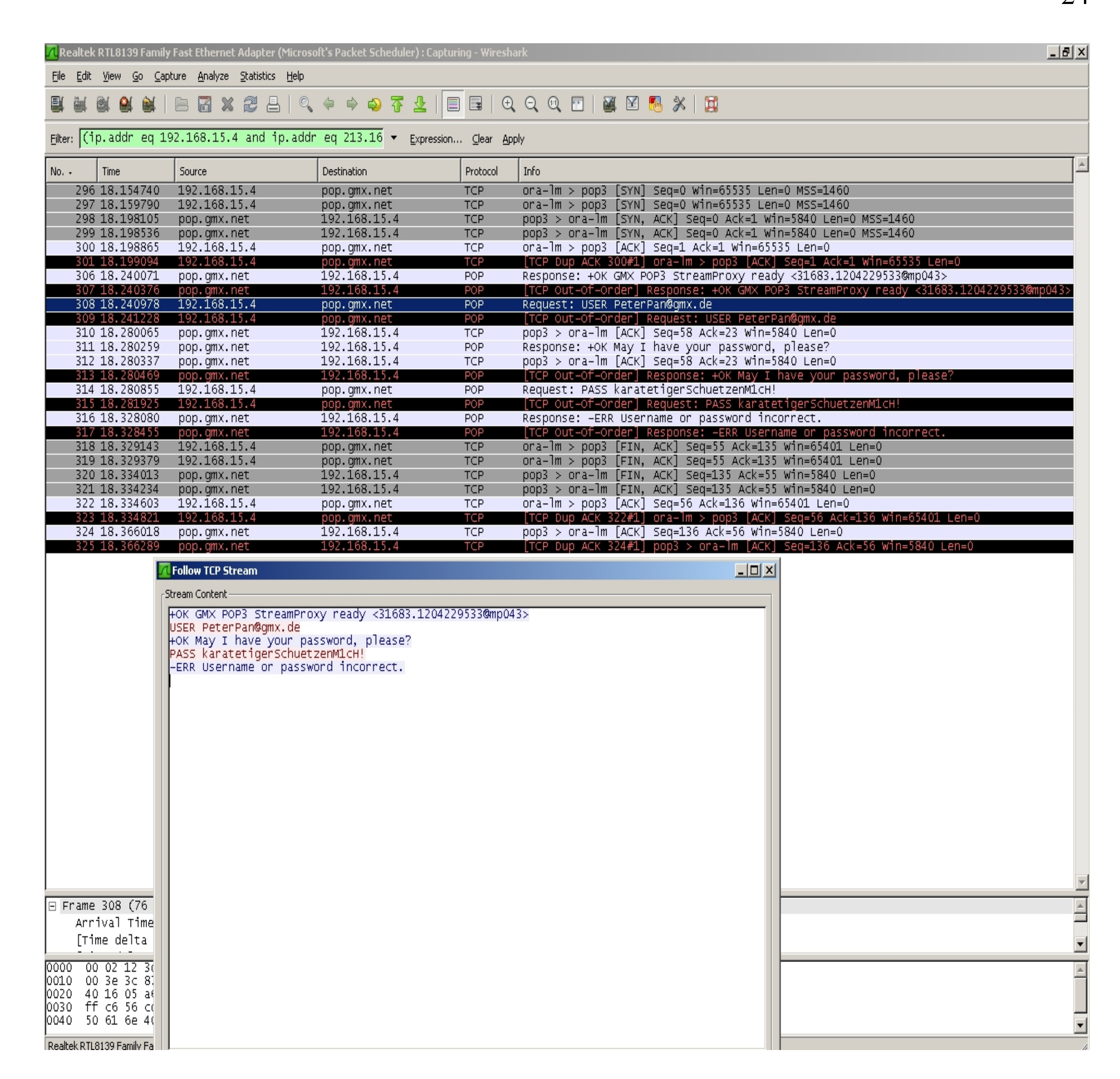


Abbildung 9: Durch ARP-Spoofing mit Wireshark abgefangener E-Mail-Login